

“华为杯”第十五届中国研究生 数学建模竞赛

题 目 机场新增卫星厅对中转旅客影响的评估方法

摘 要

机场新增卫星厅对中转旅客的换乘可能会造成一定的影响,因此通过仿真分析卫星厅的存在对机场运营的影响具有十分重要的意义。本文对新增卫星厅后的机场登机口优化分配问题进行了一定的研究。首先对数据进行预处理,将研究范围限定在 2018 年 1 月 20 日,对其中的 69 个登机口、303 个航班、863 个换乘记录及对应的 2751 名乘客进行研究,逐步建立了考虑中转旅客换乘时间的航班-登机口优化分配模型,并针对这一高维度的组合优化问题提出了对应的求解算法,求解结果表明所提算法是有效的。具体的问题解决方案如下所述。

针对问题一,本文首先以最大化被分配到固定登机口的航班最为目标函数,考虑登机口与航班的匹配约束以及登机口时间占用约束,建立了整数规划模型,通过 Gurobi 求解器求解得到拥有固定登机口的飞机最多有 256 架。进一步以约束的形式逐步减小登机口的数量,从而得到登机口数量-拥有固定登机口飞机的 Pareto 前沿,从而得到登机口的最少数量为 65。

针对问题二,本文通过引入 4 个索引描述换乘过程,然而通过传统解析法建模将会导致“维数灾”,使得问题难以求解。因此本文提出了一种多阶段邻域搜索算法,首先设计了登机口内部航班交换、登机口内部航班平移、登机口-临时机位航班交换这三种邻域搜索策略,并证明了这三种策略能遍历所有的邻域,同时设计了对应的邻域约束判断函数,从而得到有效邻域,最后证明了邻域搜索算法具有优化目标函数的梯度,通过邻域搜索算法能够在不违反约束的条件下逐步的向目标函数更优的方向移动,从而得到合理的最优解。本文以问题一中得到的登机口分配作为初始解,从而确保有 256 架航班拥有固定登机口,以中转旅客流程时间最短作为第一阶段的目标,从而得到需考虑换乘的乘客数量为 1709 名,中转旅客的流程时间为 54655 分钟。在第二阶段中,将第一阶段中求得的最短流程时间最为约束条件,优化登机口的数量,从而得到登机口的最少数量为 65。

针对问题三，本文进一步考虑中转旅客的换乘时间，将目标函数改变为最小化中转旅客的换乘紧张度，同样使用多阶段邻域搜索算法，最终得到拥有固定登机口的航班有 256 架，从而得到需考虑换乘的乘客数量为 1718 名，中转旅客的换乘紧张度为 511.06，登机口的最少数量为 66。

通过对结果的分析我们发现，卫星厅的建设可以缓解原有航站楼登机口不足的压力，但卫星厅的投运可能使游客的换乘时间增加，引起旅客换乘失败。根据计算，新建卫星厅后，旅客的平均换乘紧张度为 0.297，没有换乘失败的旅客，因此卫星厅虽然在一定程度上增加了换乘时间，但旅客的平均换乘紧张度仍是一个较低的数字，同时并没有因为卫星厅的存在导致旅客出现换乘失败的现象，因此机场新增卫星厅对中转旅客影响在可以接受的范围内，卫星厅对于机场而言是利大于弊的。

关键词：航班-登机口分配；组合优化；多目标优化邻域搜索算法；分阶段优化

目 录

1 问题重述.....	5
1.1 问题背景.....	5
1.2 待解决问题	5
2 模型假设及符号说明	6
2.1 模型假设	6
2.2 符号说明	7
3 数据预处理及问题分析.....	8
3.1 冗余数据删除处理	8
3.2 时标处理	8
3.3 逻辑变量量化处理	8
3.4 19 日到达 20 日出发的飞机处理	9
3.5 飞机起飞时间延长处理	9
3.6 问题分析	9
4 问题一的模型建立与求解	11
4.1 问题分析	11
4.2 不考虑登机口数量航班-登机口组合优化模型.....	12
4.3 考虑登机口数量及逐次逼近求 Pareto 的方法.....	13
4.4 求解结果	14
5 问题二的模型建立与求解	19
5.1 算法描述	19
5.2 算法最优性证明	23
5.3 最小登机口处理方法	23
5.4 求解结果	24
6 问题三的模型建立与求解	28
6.1 数学模型	28

6.2 求解结果	28
7. 结论	33
8 模型的优缺点与展望	35
8.1 模型优点	35
8.2 模型缺点	35
8.3 展望	35
9 参考文献	36
10 附录	37

1 问题重述

1.1 问题背景

随着旅行行业的快速发展，A 航空公司在 H 机场的现有航站楼 T 的旅客吞吐量已经达到饱和状态。为了满足未来可能增开更多航班的需求，A 航空公司计划增设 S 卫星厅，增加 S 卫星厅后，虽然可以缓解原有航站楼 T 登机口不足的压力，但是由于航站楼 T 与卫星厅 S 之间有一定的距离，势必会对中转旅客的有效衔接带来影响。不合理的航班-登机口安排计划一方面会导致大量中转旅客没有足够的通勤时间完成换乘，增加旅客换乘失败率，另一方面也可能会引起大量登机口有较低的航班使用率，不利于航空公司的运行经济性。因此，有必要对航班-登机口进行合理优化，使得更多的航班有合适登机口位置，同时以降低中转旅客的换乘失败率、提高登机口使用率^[1]。

飞机在机场廊桥（登机口）的一次停靠通常由一对航班（到达航班和出发航班，也叫“转场”）来标识。航班-登机口分配就是把这样的航班对分配到合适的登机口。所谓的中转旅客就是从到达航班换乘到由同一架或不同架飞机执行的出发航班的旅客。

航站楼 T 具有完整的国际机场航站功能，包括出发、到达、出入境和候机。卫星厅 S 是航站楼 T 的延伸，可以候机，没有出入境功能。航站楼 T 有 28 个登机口，卫星厅 S 有 41 个登机口，其配置见附件数据部分。T 和 S 之间有捷运线相通，可以快速往来运送国内、国际旅客。假定旅客无需等待，随时可以发车，单程一次需要 8 分钟。旅客流程可以按始发旅客、终到旅客和中转旅客分类规范。但由于新建卫星厅对始发旅客和终到旅客影响甚微，故本赛题只对中转旅客进行研究。

1.2 待解决问题

问题一

本题只考虑航班-登机口分配。作为分析新建卫星厅对航班影响问题的第一步，首先要建立数学优化模型，尽可能多地分配航班到合适的登机口，并且在此基础上最小化被使用登机口的数量。本问题不需要考虑中转旅客的换乘，但要求把建立的数学模型进行编程，求最优解。

问题二

考虑中转旅客最短流程时间。本问题是在问题一的基础上加入旅客换乘因素，要求最小化中转旅客的总体最短流程时间，并且在此基础上最小化被使用登机口的数量。本题不考虑旅客乘坐捷运和步行时间，但也要求编程并求最优解。

问题三

考虑中转旅客的换乘时间。如前所述，新建卫星厅对航班的最大影响是中转旅客换乘时间的可能延长。因此，数学模型最终需要考虑换乘旅客总体紧张度的最小化，并且在此基础上最小化被使用登机口的数量。本问题可以在问题二的基础

础上细化，引入旅客换乘连接变量，并把中转旅客的换乘紧张度作为目标函数的首要因素。和前面两个问题一样，本问题也要求把建立的数学模型进行编程，并求最优解。换乘紧张度定义：

$$\begin{aligned}\text{换乘紧张度} &= \frac{\text{旅客换乘时间}}{\text{航班连接时间}} \\ \text{旅客换乘时间} &= \text{最短流程时间} + \text{捷运时间} + \text{行走时间} \\ \text{航班连接时间} &= \text{后一航班出发时间} - \text{前一航班到达时间}\end{aligned}$$

2 模型假设及符号说明

2.1 模型假设

- 假设 1：各目标优先级：尽可能多的航班找到固定登机口>最小的旅客总体紧张度>尽可能少的固定登机口投入使用；
- 假设 2：本文采用离散时间步长，步长为 5 分钟；
- 假设 3：统计结果按航班数量计算；
- 假设 4：登机口使用率是指 20 号 0 点到 24 点之间的 24 小时内登机口占用时长比率，同时，使用时间不包括每架航班的 45 分钟安全间隔时间；
- 假设 5：分配到临时机位的飞机对应的航班旅客，不纳入中转旅客的最短流程时间与总体紧张度的计算之内；
- 假设 6：总体紧张度是单个紧张度之和；
- 假设 7：假设换乘失败旅客的真实换乘时间为 6 小时；

2.2 符号说明

集合	含义
X	转场飞机集合
H	登机口集合
M	航班集合
Q	航班-登机口对集合, i.e. $\{n, h\} \in \mathbf{Q}$ 表示第 n 个航班分配到了第 h 个登机口
PX	飞机属性集合 i.e. $PX(i) = (1, 0, 1)$ 表示第 i 架飞机为窄飞机, 国内到达, 国际出发
PH	登机口属性集合 i.e. $PH(h) = (1, 0, 1, 1, 2)$ 表示第 h 个登机口接收窄飞机, 国内到达, 国际出发, 位于航站楼T, 位于中部
TX	飞机停站时间集合 i.e. $TX(n) = (0, \dots, t_1, \dots, t_w, \dots, 0)$ ($t_1 = t_2 = \dots = t_w = 1$) 表示第 n 架飞机计划 t_1 时刻着落, t_w 时刻起飞
XM	飞机航班集合 i.e. $XM(n) = (i, j)$ 表示第 n 架飞机的到达航班为第 i 个航班, 出发航班为第 j 个航班
R	旅客属性集合 i.e. $R(v) = (i, j)$ 表示第 v 个旅客的到达航班为第 i 个航班, 出发航班为第 j 个航班
TS	旅客流程时间集合 i.e. $TS(g) = (t, k)$ 表示第 g 种情景下的旅客最短流程时间为 t , 需要乘坐捷运 k 次
TW	旅客行走时间集合 i.e. $TW(g) = (t)$ 表示第 g 种情景下的旅客需要行走 t 分
索引	含义
n, i, j	飞机或者航班索引
h, k, l	登机口索引
t	时刻索引
v	旅客索引
决策变量	含义
$x_{h,n}$	第 n 飞机是否分配到第 h 个登机口
use_DJK_h	第 h 个登机口是否投用
$m_{i,j,k,l}$	第 i 个航班分配到第 k 个登机口同时第 j 个航班分配到第 l 个登机口
$PT_{k,l}$	k 登机口到 l 登机口的最短流程时间
$JY_{k,l}$	k 登机口到 l 登机口的捷运时间
$W_{k,l}$	k 登机口到 l 登机口的行走时间

3 数据预处理及问题分析

赛事提供了包含 2018 年 1 月 19 日、20 日以及 21 日三天的飞机转场计划和中转旅客附件信息，按照要求，本文仅对 20 日达到或出发的航班和旅客进行分析（包含 19 日到达 20 日出发、20 日到达 20 日出发、20 日到达 21 日出发以及带有‘*’同时是在 20 日到达或出发等四种类型的所有飞机）（以本章处理后数据计算全文结果，后面不再说明）。

为了方便下文中的数据使用，需要进行一系列的数据预处理，包括冗余数据删除、时标、逻辑变量量化、特殊样本处理以及飞机起飞时间处理等。

3.1 冗余数据删除处理

从提供的全部数据中删除不在本次优化范围内的样本，只保留 20 日达到或出发的航班和旅客样本。经过统计，总计有 303 架有效飞机数据、865 有效换乘航班以及 2751 名有效旅客数据。

3.2 时标处理

飞机的降落与起飞是严格按照时间表执行的，需要对时间进行量化处理以方便下文中与时间有关约束的表示。在此定义两种不同的时间参考系：1) Time：表示与现实世界相同的真实时间（例如，Time =9:20 代表 20 日上午 9 点 20 分）；2) t：表示时间网的索引号，采用赛事推荐的 5 分钟作为时间网的划分间隔（例如，t=2 代表第二个时刻，对应的 Time 为 $5 \times 2 = 10$ ，即 00:10）。表 3.1 展示了部分飞机转场计划的时标处理结果。

表 3.1 部分飞机转场计划的时标处理结果

飞机转场 记录号	到达日期	到达时刻 (Time)	量化(t)	出发日期	出发时刻 (Time)	量化(t)
PK269	20-Jan-18	10:45	129	20-Jan-18	12:05	145
PK270	20-Jan-18	11:00	132	20-Jan-18	12:15	147

3.3 逻辑变量量化处理

每个飞机/航班以及登机口都具备一些功能属性，例如国内/国际、到达/出发、宽体机/窄体机等，飞机/航班只能分配到与之属性相吻合的登机口，因此需要将这些逻辑属性进行数值量化处理，以便以方便下文中属性匹配约束的表示。表 3.2 展示了部分飞机属性量化结果，表 3.3 展示了部分登机口属性量化结果。

表 3.2 部分飞机属性量化结果

飞机转场记 录号	到达航班	到达类型 (国内=0, 国际=1)	飞机型号、 (宽=0, 窄=1)	出发航班	出发类型 (国内=0, 国际=1)
-------------	------	-------------------------	------------------------	------	-------------------------

PK277	GN0253	0	1	GN0204	0
PK351	NV3131	1	1	NV3131	0

表 3.3 部分登机口属性量化结果

登机口	终端厅 (S=0, T=1)	区域 (N=1, C=2, S=3, E=4)	到达类型 (国际=1, 国内=0, 国际或国内=2)	出发类型 (国际=1, 国内=0, 国际或国内=2)	机体类别 (宽=0, 窄=1)
T1	1	1	1	1	1
S1	0	1	0	0	1

3.4 19 日到达 20 日出发的飞机处理

对于 19 日到达 20 日出发的每架飞机，有两种可能的登机口分配情况。1) 19 日将该飞机安排到登机口，该登机口一直被该机占用直到 20 日该机起飞；2) 19 日将该机安排到临时机位，不影响 20 日登机口的分配。本文认为可将 19 日到达 20 日出发的每架飞机处理为 20 日 $t=1$ 时刻到达并按原计划出发，再与剩余飞机共同优化。

现证明这种处理的合理性：如果优化结果将其分配到了固定登机口，该飞机从 $t=1$ 时刻占用该登机口直到 20 日该机按原计划起飞，与前文所述情况 1 影响相同；如果优化结果将其分配到了临时机位，该飞机将不占用 20 日任何时刻的任何登机口，与前文所述情况 2 影响相同，综上所述，处理后的与处理前的解空间完全相同，因此认为该处理方法是合理的。

3.5 飞机起飞时间延长处理

根据题目要求，分配在同一登机口的两飞机之间的空挡间隔时间必须大于等于 45 分钟，针对这一约束，本文通过将每台飞机的起飞时间延时 9 个时间网格长度（即 $5 \times 9 = 45$ 分钟）来轻松的解决这一约束，很容易证明这种处理方法是合理的。

由于通过已经通过数据处理保证了“同一登机口的两飞机之间的空挡间隔时间必须大于等于 45 分钟”这一约束的满足，下文的优化模型中将不再考虑这一约束，但计算结果中仍采用处理前的飞机起飞时间进行计算（下文不再说明）。

3.6 问题分析

对于每架需要转场的飞机，我们需要给它分配一个合适的登机口，以方便旅客的换乘机与离机，如果此时没有可用的登机口，则需要将这架飞机分配到临时停机坪，虽然题目中并没有提到临时停机坪的弊端，但是这必将对旅客的出行带来极大的不方便，降低旅客乘机的舒适度，因此我们应尽量避免将飞机安排到临时机位情况的发生。另一方面，假设我们有无限多的登机口可以使用，虽然可以避免安排到临时机位的情况发生，但高昂的基建费用与有限的土地空间不容许这

种假设成立，这样必然导致在较少登机口的情况下（减少登机口的建设费用与维护费用）必须牺牲部分飞机（将其安排到临时机位）。很显然，较少的登机口数量与较多的飞机分配到固定登机口之间存在矛盾，因而如何在保证经济性（最小化投运登机口数量）的同时提高旅客乘机舒适度（最大化飞机-登机口对数量）是一个有价值的数学优化问题。

同时，对于中转旅客，他们需要从着落航班所在的登机口中转到换乘航班所在的登机口，中间可能发生的出入境手续办理、搭乘捷运往返航站楼与卫星厅以及步行的所产生时间，如果旅客必要的换乘时间小于换乘航班间起降的间隔，则导致换乘失败发生，很明显，换乘失败的发生与是否合理的分配航班-登机口对是密切相关的。如何根据旅客的出行计划，合理的安排航班与登机口分配情况，以降低旅客换乘紧张度，减少换乘失败事件的发生，并在此基础上最小化登机口使用数量，同样也是机场调度中具有现实意义的数学优化问题。

我们将上述两个问题统称为登机口分配问题（Airport Gate Assignment Problem, AGAP）。由于本赛题的优化目标的优先级为：尽可能多的航班找到固定登机口>最小的旅客总体紧张度>尽可能少的固定登机口投入使用。基于该假设，对航班-登机口分配的优化（最小的旅客总体紧张度）必须是以已知有固定登机口的航班数量为前提的，这是优化目标的优先级决定的。

针对题目中的三问，本文的求解思路为：

问题一：

由于问题一不需要考虑中转旅客的换乘，因此可以暂不考虑第二个目标，直接考虑尽可能多地分配航班到合适的登机口，并且在此基础上最小化被使用登机口的数量。运用(0-1)变量对航班分配到合适的登机口进行表示，以最大化分配到登机口的航班数为目标，并以航班类型和登机口类型匹配、两架航班落到共用同一登机口的时间限制作为约束，对其进行优化求解。

问题二：

此问在第一问基础上增加了旅客的换乘环节，即将表示航班分配到合适登机口的(0-1)矩阵转换为含有起降航班以及起降登机口的四维矩阵，并以旅客换乘时间最短作为目标函数，仍以登机口匹配以及航班占用登机口的时间为约束，对其进行优化求解。根据求得在旅客最小换乘时间时占用的登机口数量作为求最小登机口时约束的上限，以最小登机口为目标函数，进一步对登机口数量进行优化，从而实现在保证换乘总时间不变的基础上，进一步减少登机口的数量。

问题三：

此问在第二问基础上继续细化旅客的换乘时间，将不同航站楼进行分区来计算行走时间，在解题思想上与第二问保持一致，目标函数更换为旅客总体紧张程度最低，约束保持与第二问一致，即可求解出旅客最小紧张程度。以在旅客最小紧张程度下的登机口数量作为求解最小登机口时的约束上限，继续以最小登机口作为目标函数，进一步优化登机口数量。

4 问题一的模型建立与求解

4.1 问题分析

问题 1 需要在不考虑中转旅客换乘的前提下,尽可能多地分配航班到合适的登机口,并且在此基础上最小化被使用登机口的数量。由于航站楼 T 与航站楼 S 最多只有 69 个可用登机口,但是 20 号需要分配的飞机数量却高达 303 架,飞机数量远远超出了登机口数量,显然,如果不进行合理的航班-登机口对分配,将会出现大量的飞机分配到临时机位的情况发生,因此需要建立数学模型进行优化,使得尽可能多的飞机有固定登机口以提高旅客乘机舒适度,并在此基础上最小化投运登机口数量以保证航空公司运营的经济性。

AGAP 问题是一个有一定约束的整数线性优化模型 (ILP),由于我们需要引入更多的辅助决策变量以保证各种约束的满足,约束的复杂程度直接决定了该模型的复杂度。AGAP 问题所可能要考虑的约束条件有很多,例如机型与登机口的匹配,飞机航班属性与登机口的匹配,旅客流程时间等。根据题目要求,问题 1 只需要考虑飞机属性与登机口属性相匹配以及分配在同一登机口的两飞机之间的空挡间隔时间必须大于等于 45 分钟这两个约束条件,模型的复杂度大大简化。因此本文拟采用通过构建合适的数学模型并通过商业求解器 (例如 Cplex、Gurobi 等) 直接求解的方法完成问题 1 的求解。

本文计划采用逐点优化逼近的方法以达到拥有固定登机口的航班-投入登机口数量的 Pareto 前沿。首先,求解在不同登机口投入数量约束下的最大化有固定登机口航班数量问题,然后从解集中找到既有最大化有固定登机口航班数量,同时投用登机口数量最小的解,也就是问题一的最终解。

具体流程如图 4.1 所示。

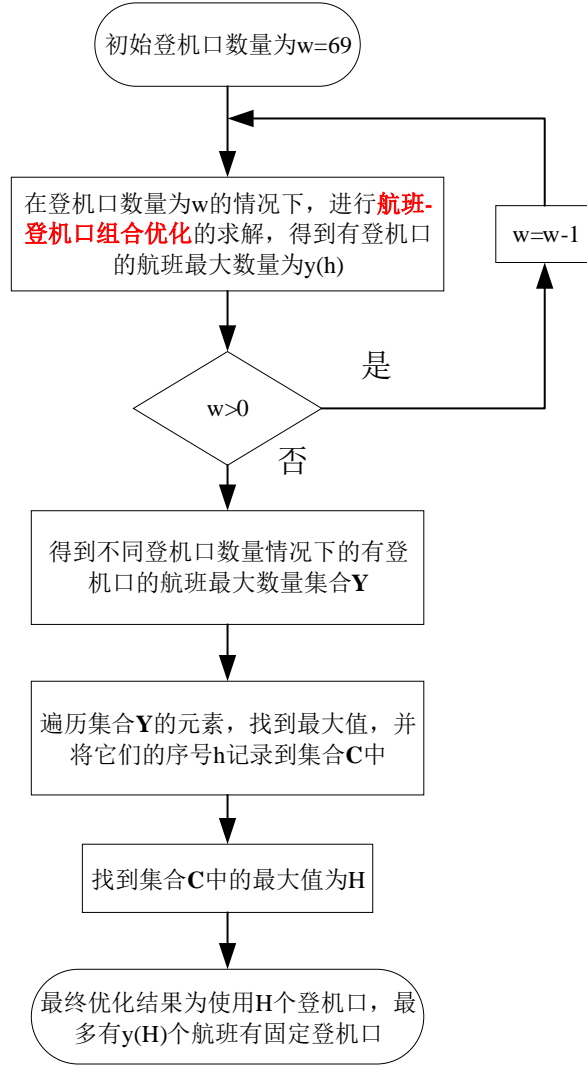


图 4.1 问题 1 优化流程图

4.2 不考虑登机口数量航班-登机口组合优化模型

由于单台飞机对应两个不同航班，同一个航班只能由一架飞机执行，因此最大化有登机口的航班数量与最大化有登机口的飞机数量是等价的。基于此，下文以最大化有登机口的飞机数量为优化目标。

首先对不考虑登机口数量的模型进行优化，其目标函数与约束如下式所示^[2]。

$$\max \sum_{h=1}^H \sum_{n=1}^N x_{h,n} \quad (4.1)$$

约束条件：

$$\sum_{h=1}^H x_{h,n} \leq 1 \quad (\forall n \in \mathbf{N}) \quad (4.2)$$

$$x_{h,n} = 0, \quad (PH(h) \neq PX(n)) \quad (4.3)$$

$$\sum_{n=1}^N x_{h,n} TX(n) \leq 1 \quad (\forall h \in H) \quad (4.4)$$

$$x_{h,n} \in \{0,1\}, y_h \in \{0,1\} \quad (h \in H, n \in N) \quad (4.5)$$

目标函数为最大化分配到登机口的飞机数量， $x_{h,n}$ 表示第 n 架飞机是否分配到第 h 个登机口。式(4.2)表示每架飞机最多只能分配到 1 个登机口。式(4.3)表示只有在登机口 h 属性与飞机 n 属性匹配的情况下，登机口 h 才有可能被飞机 n 使用，此时 x 等于 1，否则为 0。式(4.4)表示任意登机口在任意时刻其所停留的飞机不能超过 1 架。

从上述的目标函数与约束条件看，其是含有 (0-1) 变量的混合整数线性规划的问题，对于含有 N 架飞机与 H 个登机口，只需求解 $N \times H + H$ 个 0-1 决策变量，时间复杂度与空间复杂度较低，求解难度较小，可以运用 GUROBI 等求解器进行求解。随着飞机与登机口的数量的增加，时间复杂度与计算复杂度的增加速度在可以允许的范围内。

4.3 考虑登机口数量及逐次逼近求 Pareto 的方法

题目中要求不仅要把更多的航班分配到登机口，而且要使得登机口数量最小，因此在 4.1 求出能够被分配航班数的基础上，增加投运登机口数量约束，运用 pareto 逐次逼近，从而求得投运登机口最小数量。

对于投入登机口数量为 w 下的组合优化问题，其完整的数学模型如下所示：
目标函数：

$$\max \sum_{h=1}^H \sum_{n=1}^N x_{h,n} \quad (4.6)$$

约束条件：

$$x_{h,n} \leq y_h \quad (\forall n \in \mathbf{N}, \forall h \in \mathbf{H}) \quad (4.7)$$

$$\sum_{h=1}^H x_{h,n} \leq 1 \quad (\forall n \in \mathbf{N}) \quad (4.8)$$

$$x_{h,n} = 0, \quad (PH(h) \neq PX(n)) \quad (4.9)$$

$$\sum_{n=1}^N x_{h,n} TX(n) \leq 1 \quad (\forall h \in H) \quad (4.10)$$

$$\sum_{h=1}^H y_h \leq w \quad (4.11)$$

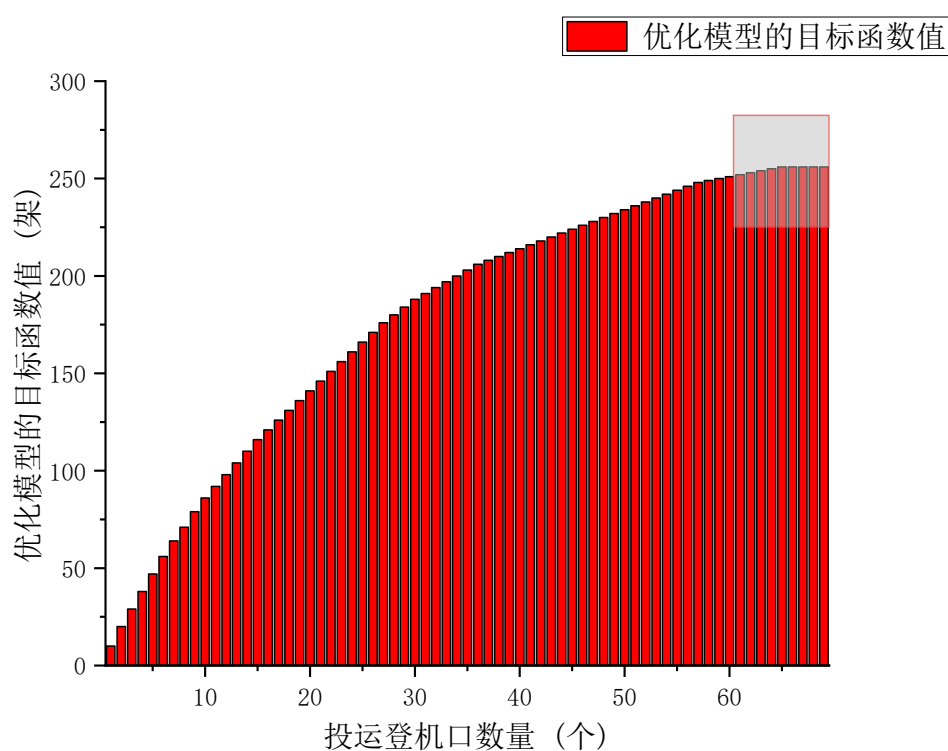
$$x_{h,n} \in \{0,1\}, y_h \in \{0,1\} \quad (h \in H, n \in N) \quad (4.12)$$

目标函数为最大化分配到到登机口的飞机数量， $x_{h,n}$ 表示第 n 架飞机是否分配到第 h 个登机口。式(4.7)表示飞机只能分配到投入使用的登机口，式(4.11)限制了本次优化的投运登机口数量最多为 w 个。

上述问题仍为整数规划问题，本文采用 Matlab 中的 Yalmip 工具箱^[3]建模并调用 GUROBI 求解器^[4]进行求解。

4.4 求解结果

4.4.1 不同登机口投用数量下的最大优化结果



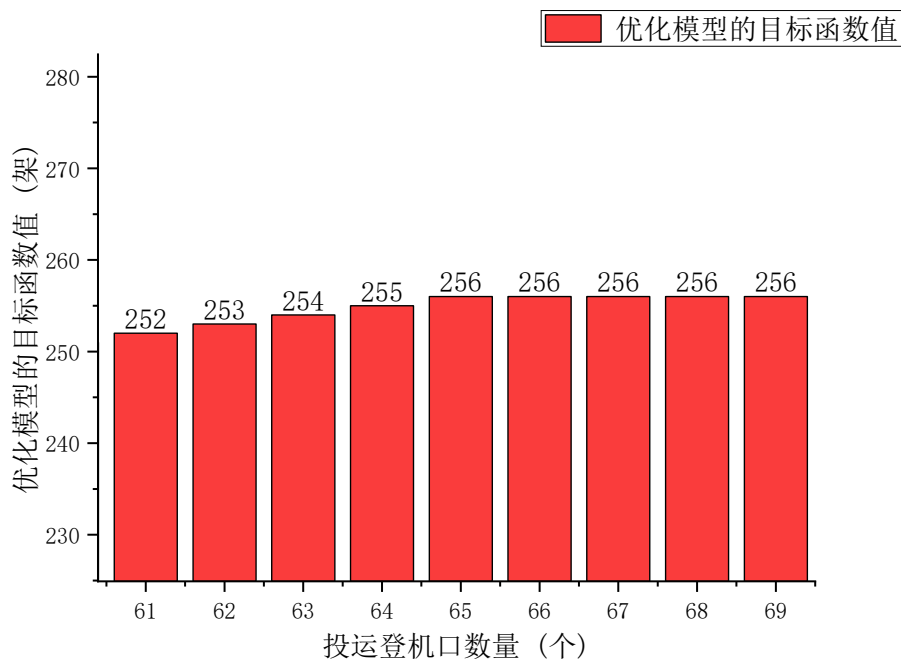


图 4.2 不同投入登机口数量下的最大有登机口飞机数量

从图 4.2 可以看出，随着投运登机口数量的减少，能分配到登机口的最大飞机数量明显减少，这是显而易见的。运用 Pareto 逐次逼近的方法，能够发现当登机口数量为 65-69 时，优化得到的最大航班-登机口对数均为 256，由于需要“尽可能多地分配航班到合适的登机口，并且在此基础上最小化被使用登机口的数量”，很显然，最优解为使用 65 个登机口，安排 256 个航班-登机口对。

4.4.2 最优情况下，成功分配的飞机情况

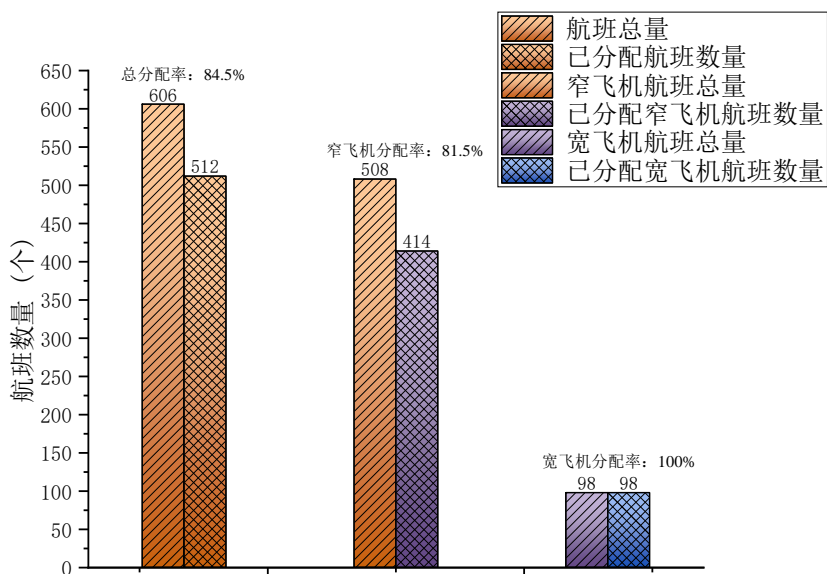


图 4.3 最优分配情况下各机型飞机分配情况

从图 4.3 可以看出，在投运 65 个登机口的情况下，有 512 个航班（84.5%）

分配到了有效的登机口，其中窄飞机航班 414 个，占窄飞机航班总量的 81.5%，宽飞机航班 98 个，占宽飞机航班的 100%。

4.4.3 最优情况下，登机口使用情况与平均使用率

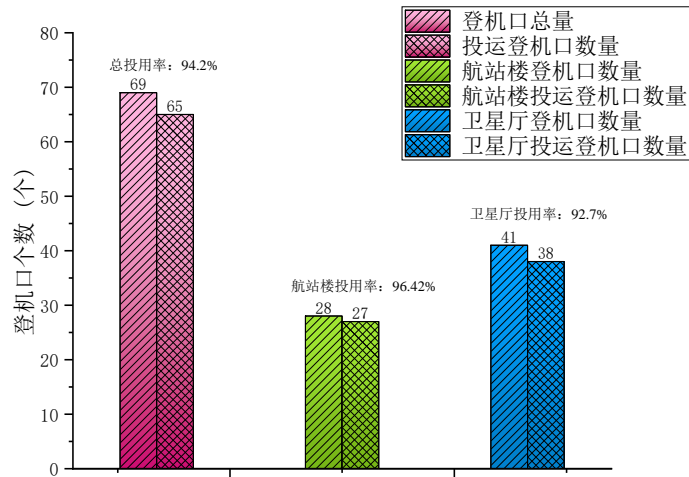


图 4.4 最优分配情况下登机口使用情况

从图 4.4 可以看出，为了将尽可能多的飞机分配到合适的登机口，需要投运很多登机口（只禁了 4 个登机口）。

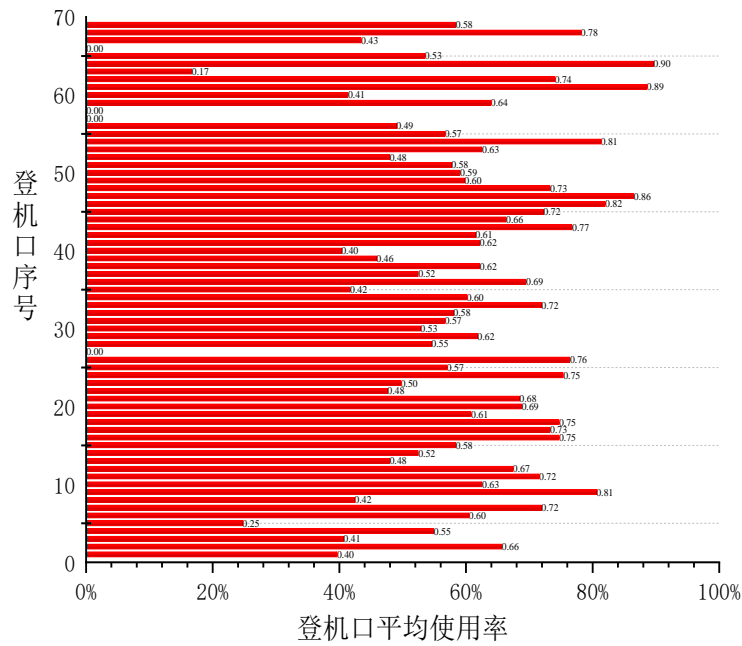


图 4.5 各登机口平均使用率

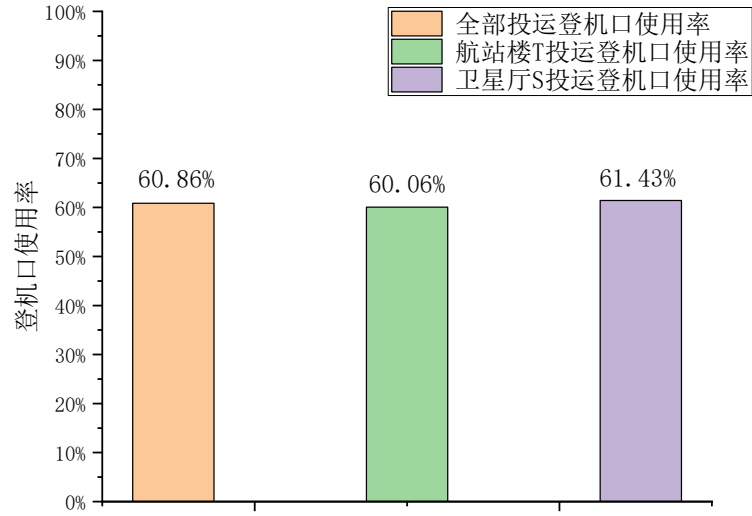


图 4.6 航站楼、卫星厅投运登机口平均使用率

定义登机口 h 的平均使用率为 $y(h)$

$$y(h) = \frac{\text{全部分配到该登机口的飞机经停时间段总和}}{288} \quad (4.13)$$

其中，航站楼与卫星厅的平均使用率为其对应登机口的平均使用率的平均值。

图 4.5 显示了不同登机口的平均使用率，图 4.6 显示了航站楼、卫星厅投运登机口的平均使用率，平均使用率反应该登机口的使用情况。为了最小化被使用登机口的数量，显然需要提高每个登机口的使用率，从图 4.5 可以看出。各个登机口的使用率均较高。

4.4.4 最优情况下，全机场航班-登机口时间流图

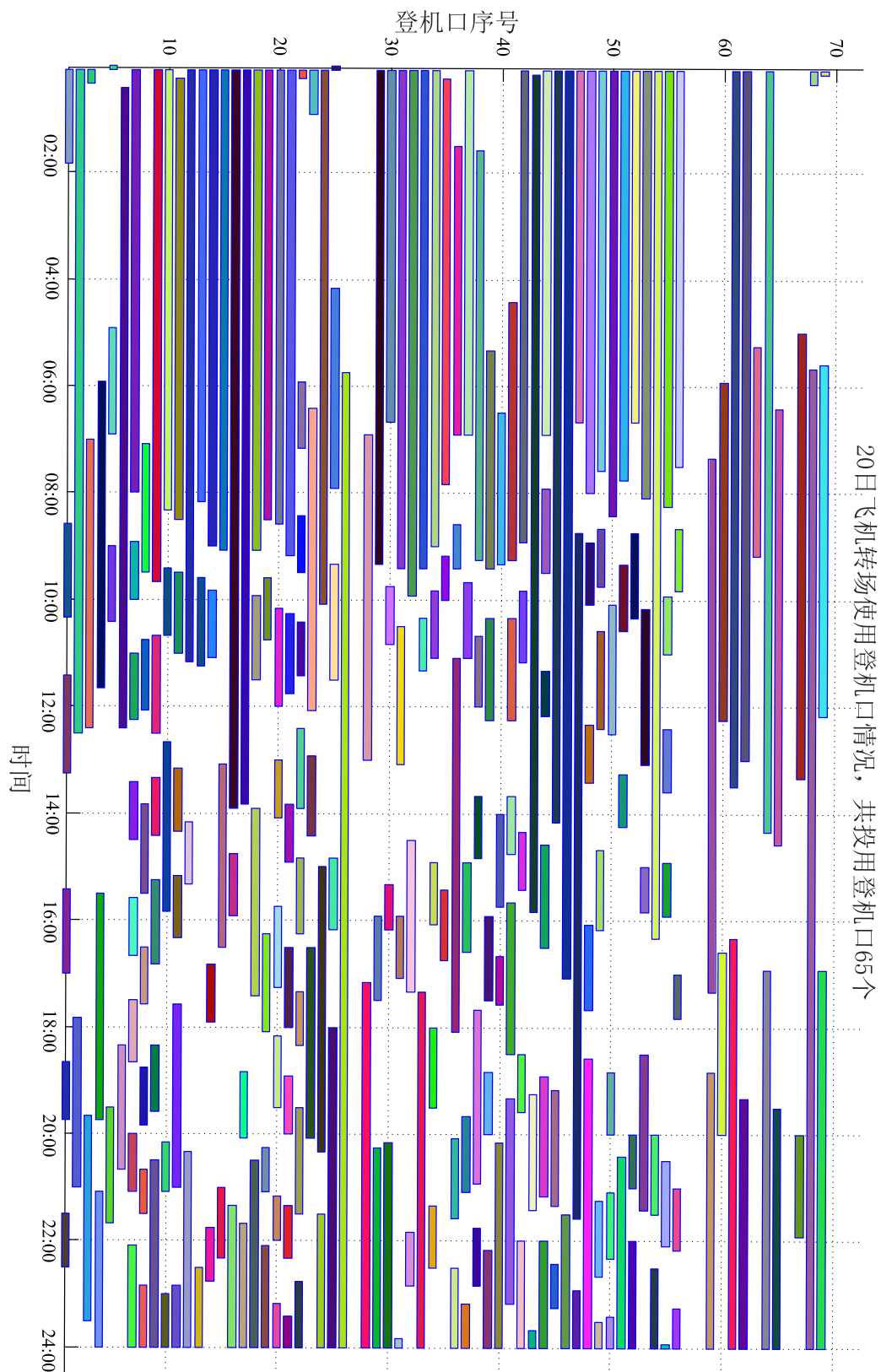


图 4.7 20 日全天各飞机起降时间流图

图 4.7 为有登机口的 256 架飞机按照时间使用登机口情况，可直观展示全天的调度方案。

5 问题二的模型建立与求解

5.1 算法描述

第一问中运用 X 矩阵将航班于登机口联系在一起，本问中增加了换乘环节，若继续沿用第一问解析法思想，则矩阵中应将包含有起降飞机及其对应登机口信息，约束条件仍为第一问中的约束，目标函数变更为换乘时间最短。如式 (5.1)~(5.6) 所示^[2]。

目标函数：

$$\sum_{i=1}^H \sum_{j=1}^H \sum_{k=1}^N \sum_{l=1}^N m_{i,j,k,l} w_{k,l} f_{i,j} \quad (5.1)$$

约束条件：

$$\sum_{k=1}^N x_{i,k} = 1 \quad (\forall i \in \mathbf{H}) \quad (5.2)$$

$$m_{i,j,k,l} \leq x_{i,k} \quad (\forall i, j \in \mathbf{H} \quad \forall k, l \in \mathbf{N}) \quad (5.3)$$

$$m_{i,j,k,l} \leq x_{j,l} \quad (\forall i, j \in \mathbf{H} \quad \forall k, l \in \mathbf{N}) \quad (5.4)$$

$$x_{i,k} + x_{j,l} - 1 \leq m_{i,j,k,l} \quad (\forall i, j \in \mathbf{H} \quad \forall k, l \in \mathbf{N}) \quad (5.5)$$

$$T_{i,j} + T_{j,i} \geq m_{i,j,k,k} \quad (\forall i, j \in \mathbf{N} \quad \forall k \in \mathbf{H}) \quad (5.6)$$

式 (5.2) 约束体现了保证 i 航班停在 k 登机口，式 (5.3)~(5.5) 约束表示只有 i 航班落在 k 登机口，j 航班落在 l 登机口，此时 $m_{i,j,k,l}$ 才有意义，式 (5.6) 约束表明了一个登机口不能同时被两架不同的航班占用。

可以看出， $m_{i,j,k,l}$ 矩阵维度为 $O(n^2 h^2)$ ，共含有约 4.37 亿的元素，矩阵空间复杂度极高。若继续沿用第一问解析方法中的分支定界法对其进行求解，时间复杂度可达 $O(2^n)$ 。因此从空间复杂度和时间复杂度来看，运用解析法思想在换乘航班数以及登机口数较大的场景下不再适用。

本问题借鉴 H Ding 提出的 New neighbourhood search methods 思想^[5-6]，运用启发式邻域搜索策略对换乘旅客的总体最小流程时间的目标函数进行求解，邻域搜索的目的是在第一问求出的最小登机口的基础上，将某两个登机口中的单个或多个航班进行对换，从而改变旅客换乘时间的方法。将邻域搜索分为三种策略，如下所示。

(1) 插入类移动：如图 5.1 中所示，将登机口 l 中的 flightC 换到登机口 k 中的空闲时间，在满足航班口匹配以及时间约束条件下，此种方案能够改变旅

客的换乘时间。

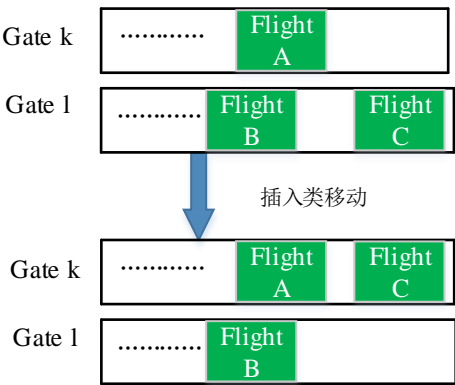


图 5.1 插入类移动示意图

(2) 间隔类交换移动：如图 5.2 中所示，k 登机口中以 flightA 落地时间为起始时间和以 flightB 起飞时间为终止时间内的所有航班，与 1 登机口中以 flightC 落地时间为起始时间和以 flightD 起飞时间为终止时间内的所有航班整体进行替换。这种替换方式包含了不同登机口单架或多架飞机的整体交换，此类交换方法也能够对换成时间产生影响。

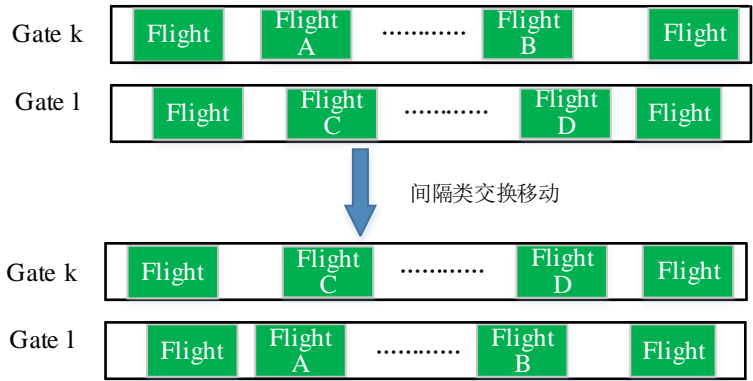


图 5.2 间隔类交换移动示意图

(3) 与临时停机坪的交换移动：根据第一问中结果可以看出，为满足所有的约束，临时停机坪是必不可少的一环。题目中对临时停机坪不考虑换乘时间，因此与登机口的航班进行交换后必定会对换乘时间产生影响。

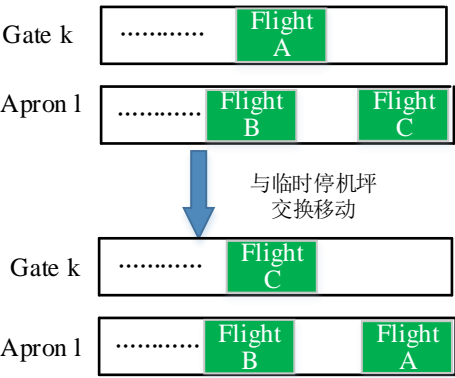
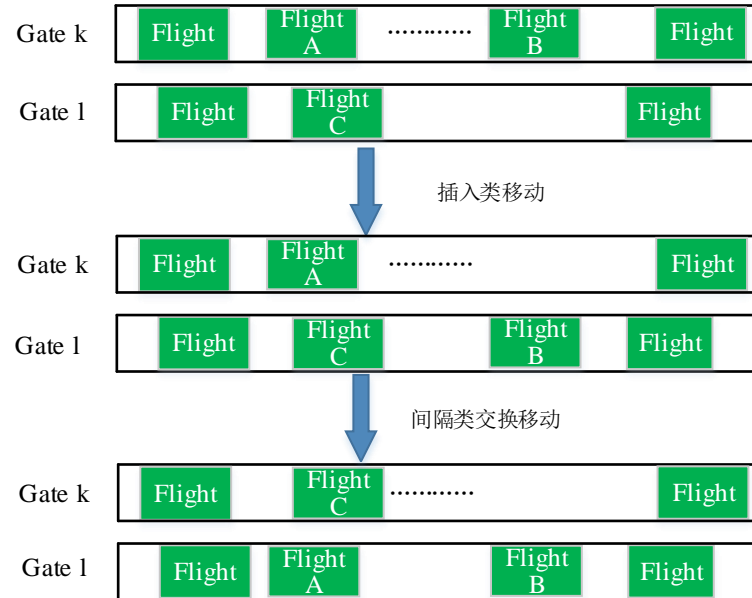


图 5.3 与临时停机坪交换移动示意图

策略 2 中对连续的时间间隔进行对换，若对不连续航班之间的对换也可以通过策略 1 和策略 2 的组合实现，如图 5.4 所示，k 登机口中的 flightA 和 flightB

起飞中间时刻内包含一定数量的航班，第一步运用插入类策略，将 flightB 插入 1 登机口空白时间内，其次运用间隔类交换移动，对 flightA 和 flightC 两架飞机进行对换，从而实现了非连续类航班间的交换。



如图 5.4 非连续类航班间交换

本题目标目标函数与约束条件如下式 (5.7) ~ (5.9) 所示。

目标函数：

$$F=f_1(x) \quad (5.7)$$

式中， f_1 表示由航班-登机口分配数据求取总换乘时间的函数。

约束条件：

$$x_{h,n} = 0 \quad (PH(h) \neq PX(n)) \quad (5.8)$$

$$\sum_{n=1}^N x_{h,n} TX(n) \leq 1 \quad (\forall h \in H) \quad (5.9)$$

为方便流程图中表述，将邻域交换策略中的插入类移动称为算法 1，间隔类交换移动称为算法 2，与临时停机坪的交换移动称为算法 3，将式 (5.8) 和式 (5.9) 的约束称为算法 4 和算法 5，将换乘时间最少的目标函数称为算法 6。优化的流程图如图 5.5 所示。

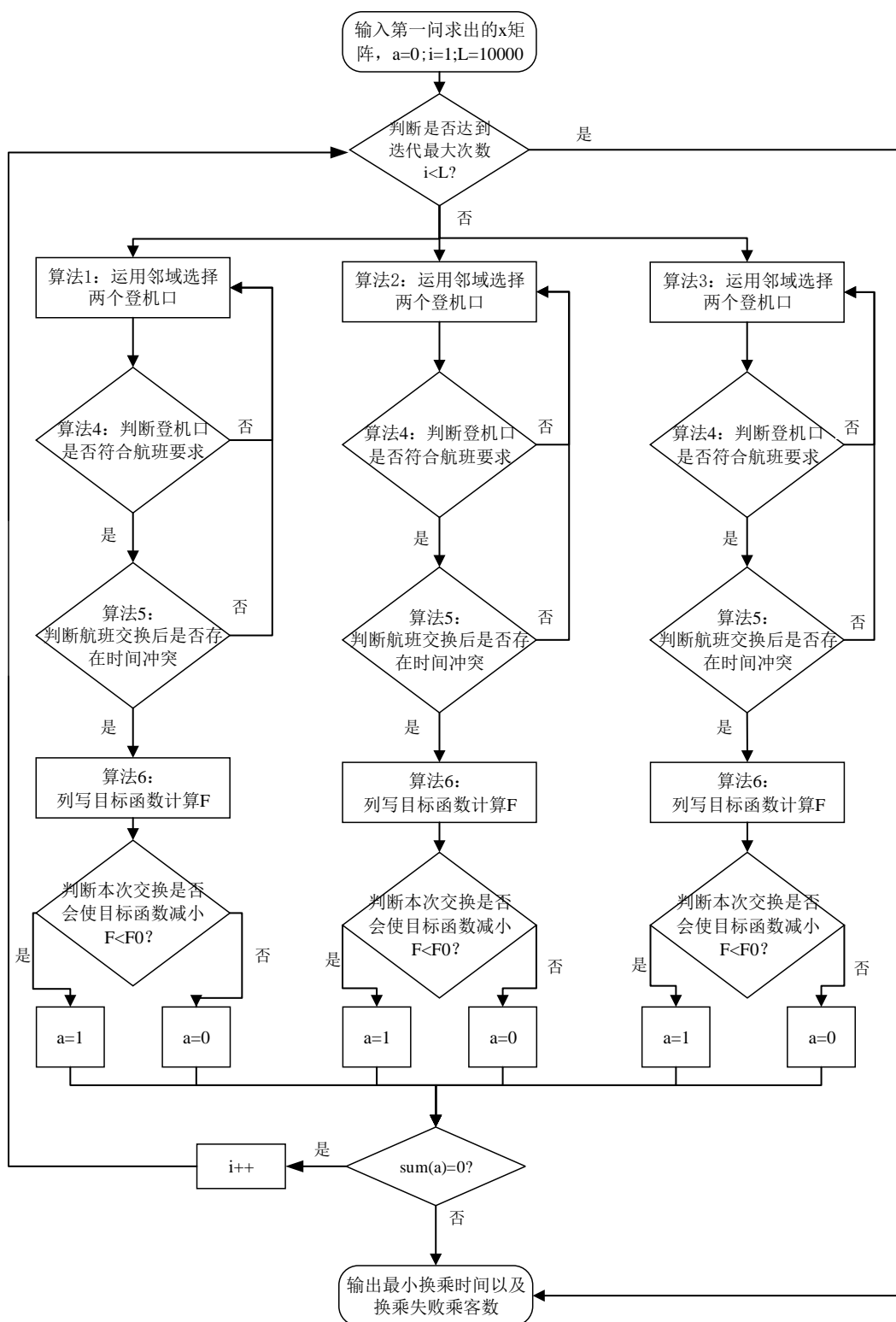


图 5.5 运用启发式邻域搜索法对最小换乘时间优化的流程图

5.2 算法最优性证明

三种策略均为单步优化，容易证明三种策略及其组合可以包含所有的邻域情况。以下证明邻域搜索算法具有最优解的梯度。

x 为目标函数 F 的当前解集，将此解集对应的目标函数记为 $F(x)$ 。若 $\exists x^*$ 对应的目标函数为 $F(x^*)$ ，假设 $F(x^*)$ 为全局最优，记为 $F(x^*) > F(x)$ 。

m 代表单步优化的次数，若 $\exists m$ 能够使得 x 解集经过 m 个单步过程过渡为 x^* 解集，即 $x \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_m \rightarrow x^*$ ，则对应的目标函数也有

$F(x) \rightarrow F(x_1) \rightarrow F(x_2) \rightarrow \cdots \rightarrow F(x_m) \rightarrow F(x^*)$ 。此时 $f(x^*)$ 能够表达为如式 (5.10) 的形式。

$$F(x^*) = F(x^*) - F(x_m) + F(x_m) - F(x_{m-1}) + \cdots + F(x_2) - F(x_1) + F(x_1) - F(x) + F(x) \quad (5.10)$$

若计 $\Delta F(x_m) = F(x_m) - F(x_{m-1})$ 和 $\Delta F(x) = F(x^*) - F(x)$ ，则式(5.7)可以变形为式(5.11)。

$$\Delta F(x) = (F(x^*) - F(x_m)) + \Delta F(x_m) + \cdots + \Delta F(x_2) + (F(x_1) - F(x)) \quad (5.11)$$

由于 $F(x^*)$ 优于 $F(x)$ ，即 $\Delta F(x) > 0$ ，因此等式右边中至少存在一组两两邻域的差值大于 0，即在 x 和 x^* 解集空间中存在单步优化能够使得目标函数更优，当 $x = x^*$ 时，即不存在单步优化能够使得目标函数更优，此时记 x 为目标函数的全局最优。综上，单步最优有机会实现全局最优，且本文提出的三种航班对换策略均是在符合所有约束条件下进行的，因此三种策略的决策顺序不会对最终的全局最优结果产生影响。但邻域搜索算法的解质量依赖于初始解空间，若初始解空间没有足够广的邻域，可能导致向全局最优移动的梯度中断，即算法陷入了局部最优。

此算法为多项式复杂度的算法，根据策略可以在有限的时间内进行求解，并且每一步的优化均满足所有的约束条件，此算法的时间和空间维度均比解析法大大降低，且其单步优化的结果最终能够收敛至最优解，所以此算法能够在较短的时间内得出准确的最优解。

5.3 最小登机口处理方法

已经求出旅客总体换乘流程最短时间，若想在此基础上进一步最小化登机口数量，则需要保证在不增加总换乘时间条件下，以运用临时停机坪数量最小为目标函数，运用邻域搜索的方法运用上述所说的三种策略进行计算，从而得出最小化登机口的数量。因此本文将得到的最小换乘时间作为约束条件，只有当搜索到的邻域的换乘时间等于已得到的最小换乘，才能将其视为有效邻域，并将登机口的数量作为目标函数，作为第二阶段邻域搜索算法。因此可以在保证换乘时间不变的前提下得到最小的登机口数量。

5.4 求解结果

5.4.1 算法收敛曲线

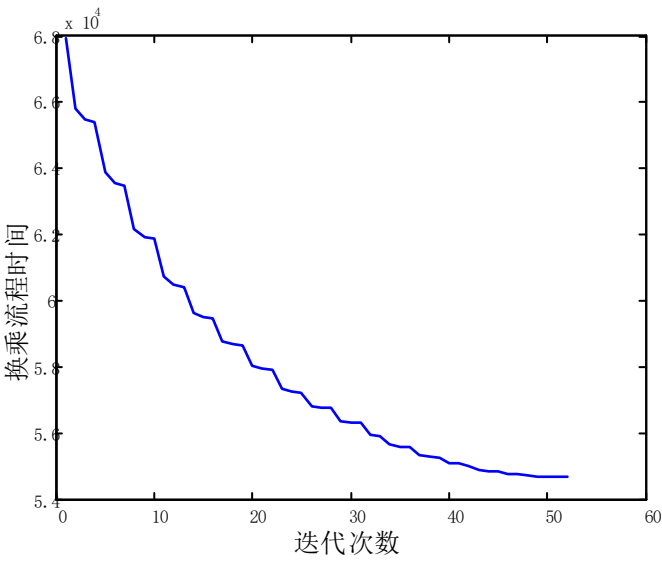


图 5.6 目标函数与迭代次数的算法收敛曲线

邻域搜索算法在经过 52 次迭代，912.7 秒的 CPU 计算时间后收敛至最优，得到的最优换乘流程时间为 54655 分钟，收敛过程如上图所示。

5.4.2 最优情况下，成功分配的飞机情况

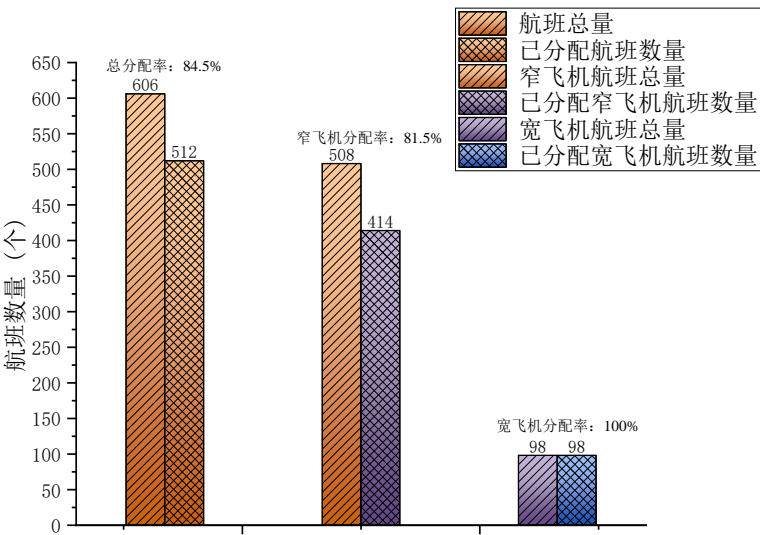


图 5.7 各机型航班分配情况

如上图所示，在最小换乘流程时间下，所有的宽飞机均被分配到了固定的登机口，而窄飞机的分配率只有 81.5%。

5.4.3 最优情况下，登机口使用情况与平均使用率

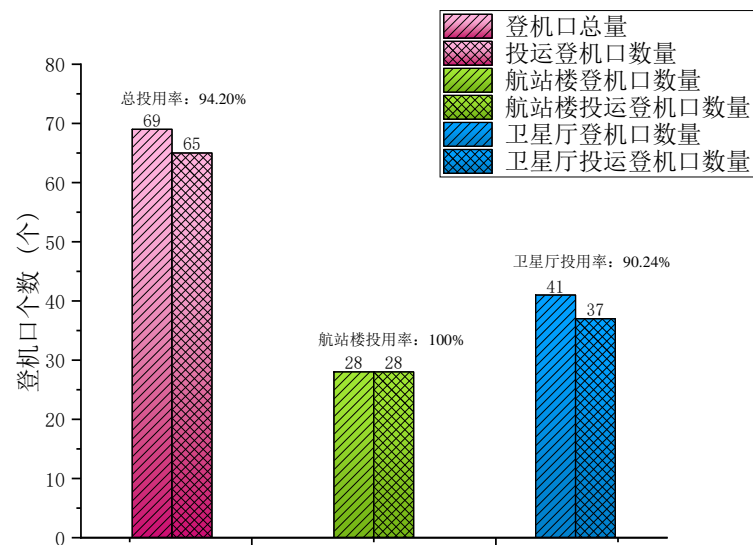


图 5.8 登机口使用情况

优化得到的最小登机口数量为 65，其中航站楼中的 28 个登机口均被利用，而卫星厅的登机口利用率为 90.24%。

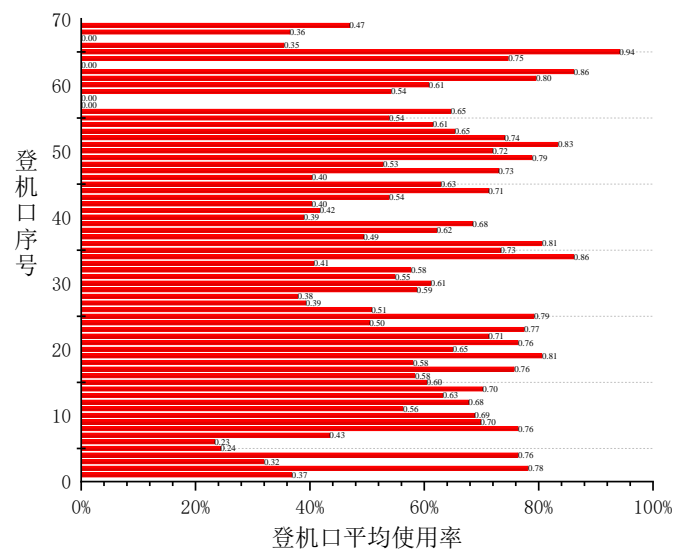


图 5.9 各登机口平均使用率

从图中可以看出，登机口的平均使用率最少为 23%，绝大多数的登机口利用率超过了 50%，可见登机口的利用率相对较高。

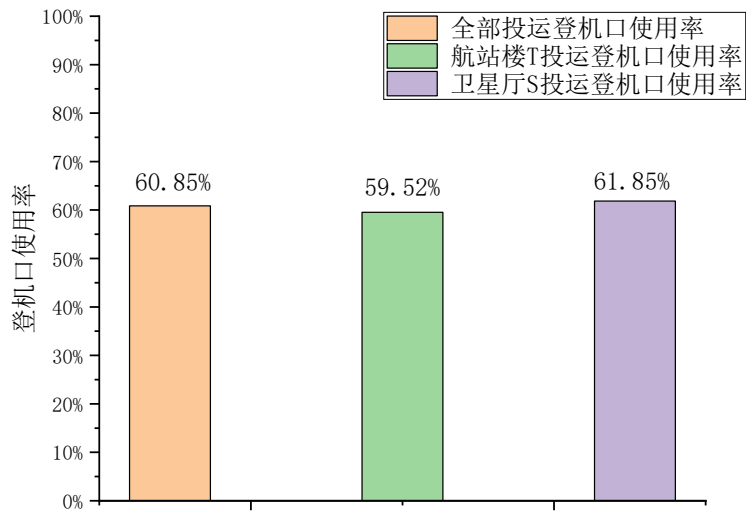
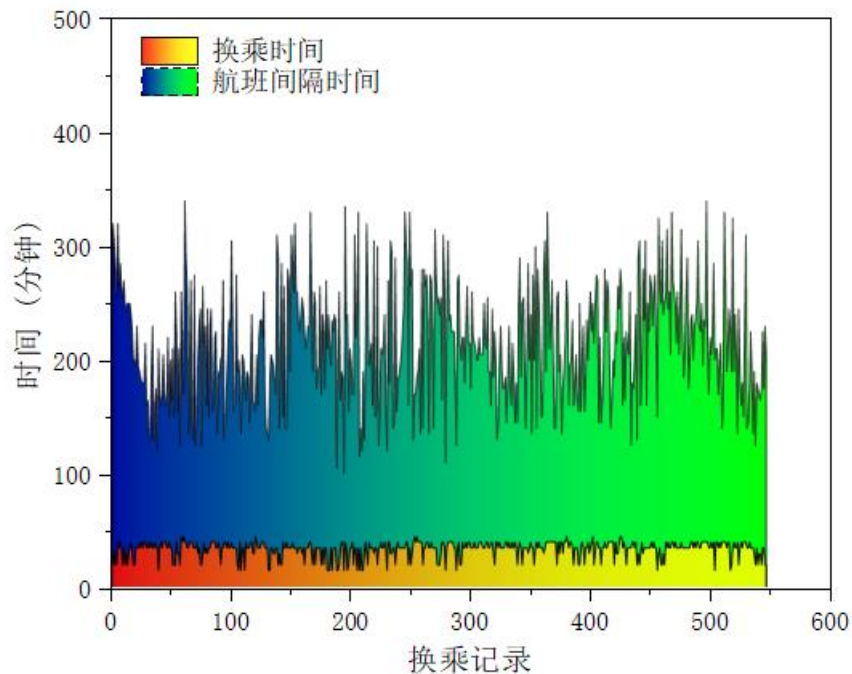


图 5.10 航站楼、卫星厅投运登机口平均使用率

从图中可以看出，登机口的平均使用率为 60.85%，而航站楼登机口的使用率为 59.52%，卫星厅登机口的使用率为 61.85%，高于航站楼登机口，可见卫星厅的登机口使用效率较高。

5.4.4 最优情况下，航班换乘时间



5.11 航班换乘时间与航班间隔时间

从图中可以看出换乘航班之间的时间间隔远比换乘过程时间长，因此没有旅客出现延误现象。

5.4.5 最优情况下，全机场航班-登机口时间流图

本文将 69 个登机口一天中被占用的形式以甘特图的形式表示出，能够直观的观测航班-登机口分配情况，如下图所示。

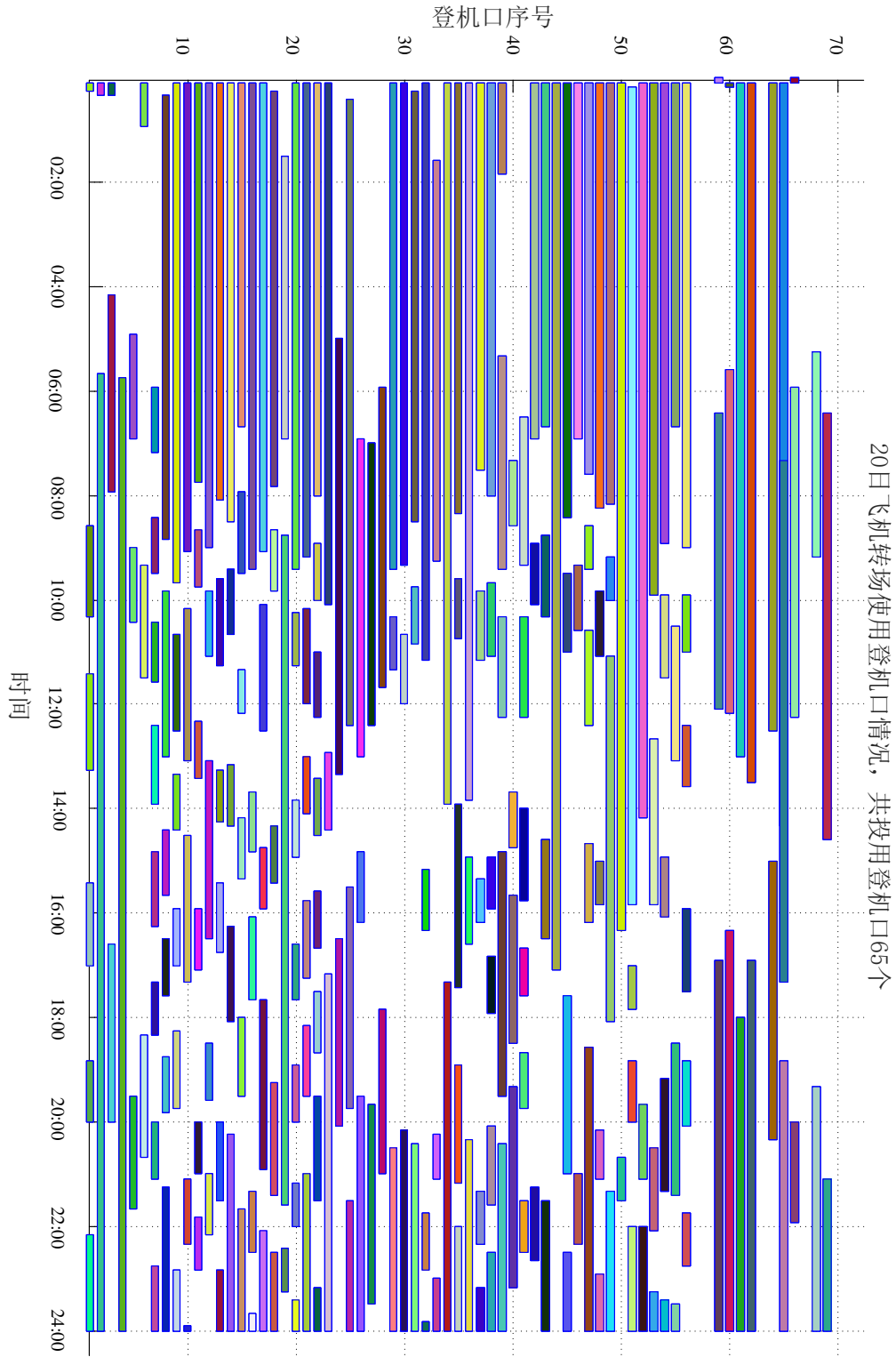


图 5.11 20 日全天各飞机起降时间流图

6 问题三的模型建立与求解

6.1 数学模型

第三问仍是运用邻域搜索法对各登机口的航班进行调换,在第二问只考虑最短换乘流程基础上考虑了捷运时间和行走时间,因此只需要对其目标函数进行修改,并且对旅客换乘时间矩阵进一步细化,则可以得出结果。

目标函数及约束条件如下式所示。

目标函数:

$$F = f_2(x) \quad (6.1)$$

式中, f_2 表示由航班-登机口分配数据求取换旅客乘紧张度的函数。

约束条件:

$$x_{h,n} = 0 \quad (PH(h) \neq PX(n)) \quad (6.2)$$

$$\sum_{n=1}^N x_{h,n} TX(n) \leq 1 \quad (\forall h \in H) \quad (6.3)$$

6.2 求解结果

6.2.1 算法收敛曲线

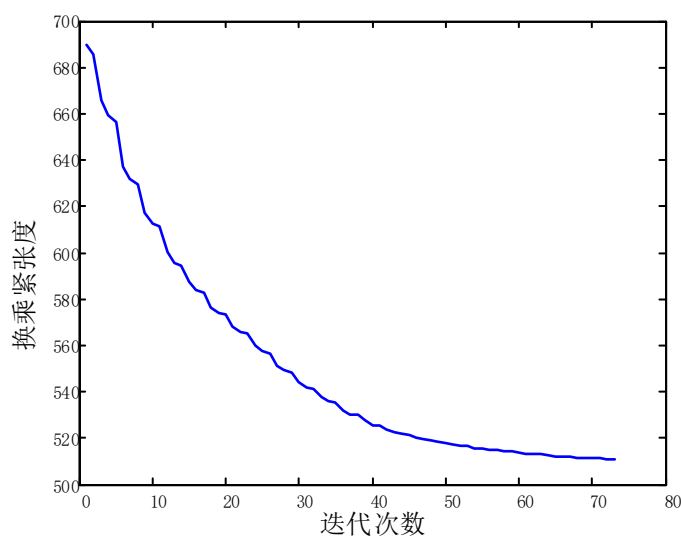


图 6.1 目标函数与迭代次数的算法收敛曲线
邻域搜索算法在经过 73 次迭代, 1527.4 秒的 CPU 计算时间后收敛至最优,

得到的最优换乘紧张程度为 511.06，收敛过程如上图所示。

6.2.2 最优情况下，成功分配的飞机情况

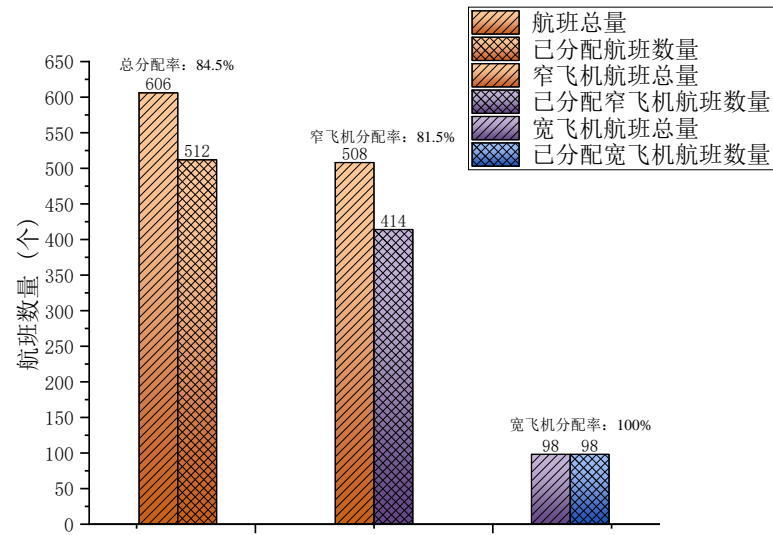


图 6.2 各机型航班分配情况

如上图所示，在最小换乘流程时间下，所有的宽飞机均被分配到了固定的登机口，而窄飞机的分配率只有 81.5%。

6.2.3 最优情况下，登机口使用情况与平均使用率

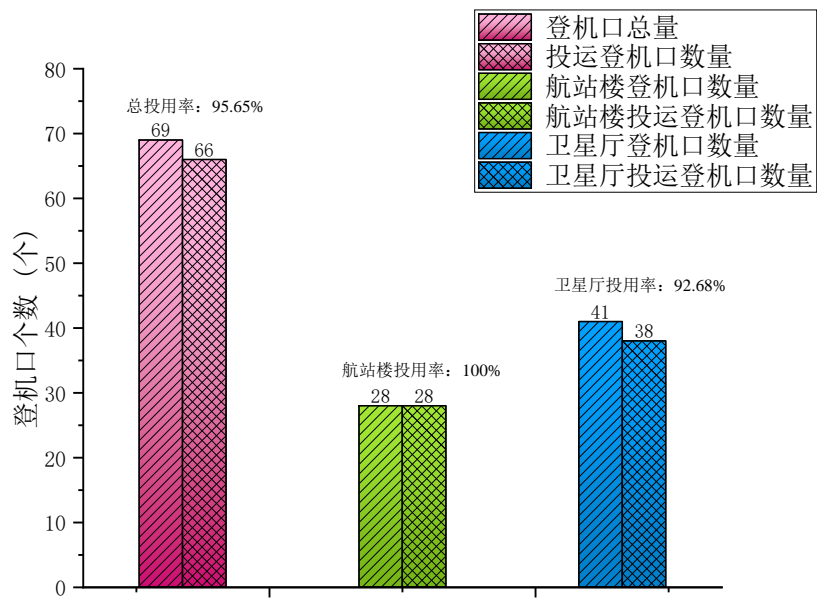


图 6.3 登机口使用情况

优化得到的最小登机口数量为 66，其中航站楼中的 28 个登机口均被利用，而卫星厅的登机口利用率为 92.68%。

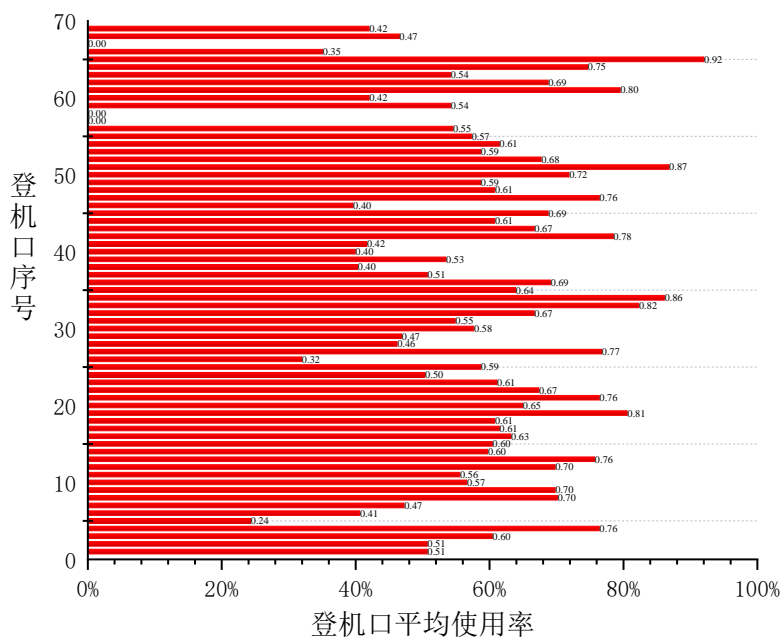


图 6.4 各登机口平均使用率

从图中可以看出，登机口的平均使用率最少为 24%，绝大多数的登机口利用率超过了 50%，可见登机口的利用率相对较高。

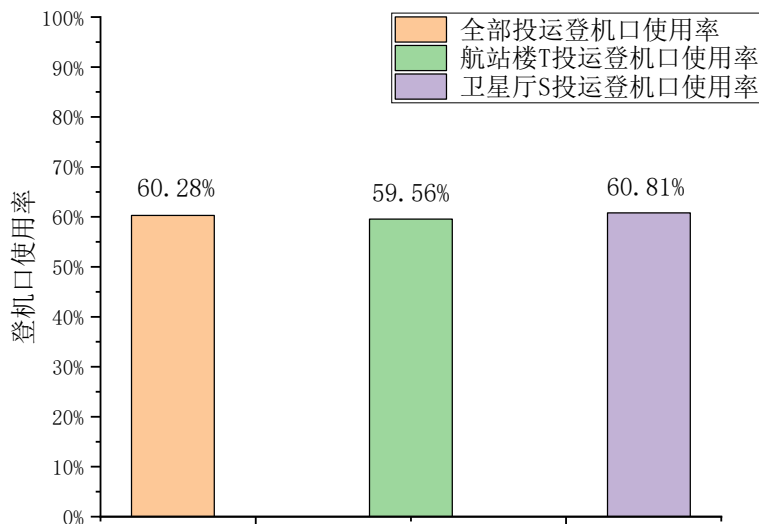


图 6.5 航站楼、卫星厅投运登机口平均使用率

从图中可以看出，登机口的平均使用率为 60.28%，而航站楼登机口的使用率为 59.56%，卫星厅登机口的使用率为 60.81%，高于航站楼登机口，可见卫星厅的登机口使用效率较高。

6.2.4 最优情况下，航班换乘时间

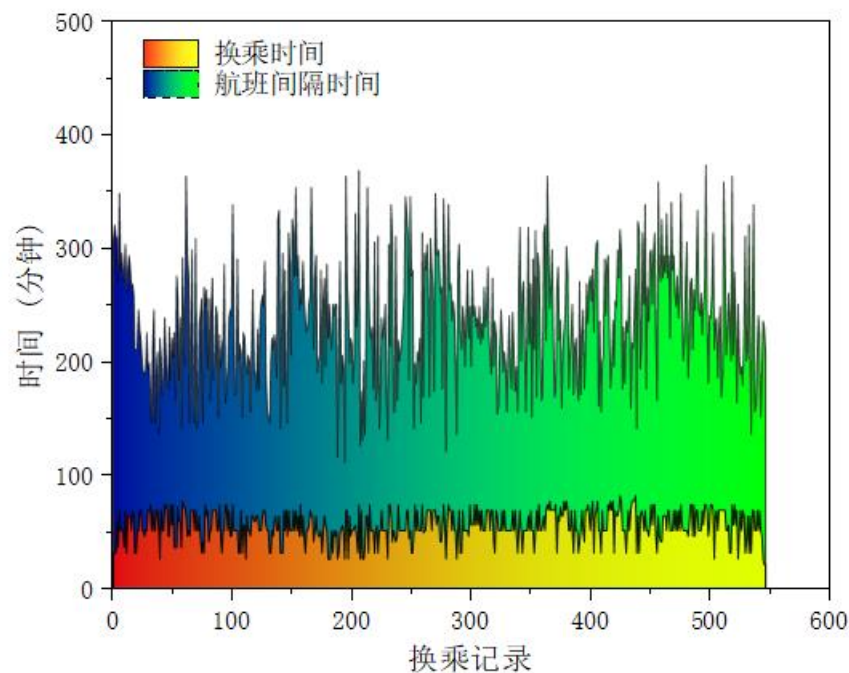


图 6.6 航班换乘时间与航班间隔时间

从图中可以看出换乘航班之间的时间间隔均比换乘过程时间长，因此没有旅客出现延误现象，同时绝大多数换乘航班给旅客换乘时间留了很大的裕度，因此旅客的平均换乘紧张度只有 0.297，大多数旅客不会因为换乘产生焦虑心理。

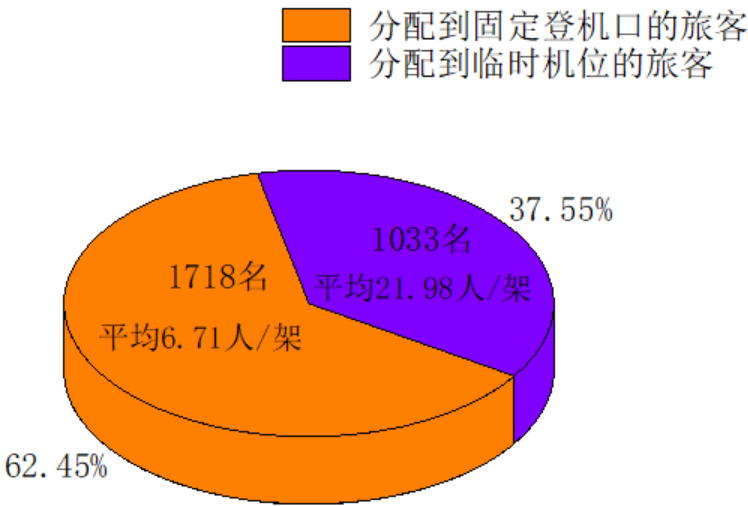


图 6.7 登机口与临时机位旅客分布

固定登机口上的飞机有 256 架，而临时登机口上的飞机只有 47 架，然而临时登机口上的飞机的平均乘客数为 21.98，而固定登机口上的飞机的平均乘客仅为 6.71。可见为了达到游客换乘紧张度最小化的优化目标，算法会优先分配拥有更多旅客数的航班至临时登机口，这与现实的情况有极大的不符。分析其原因为没有考虑分配到临时登机口航班对应游客的换乘紧张度，导致使用临时登机口的边际成本为 0，使得优化程序尽可能多的向此方向逼近。

6.2.5 最优情况下，全机场航班-登机口时间流图

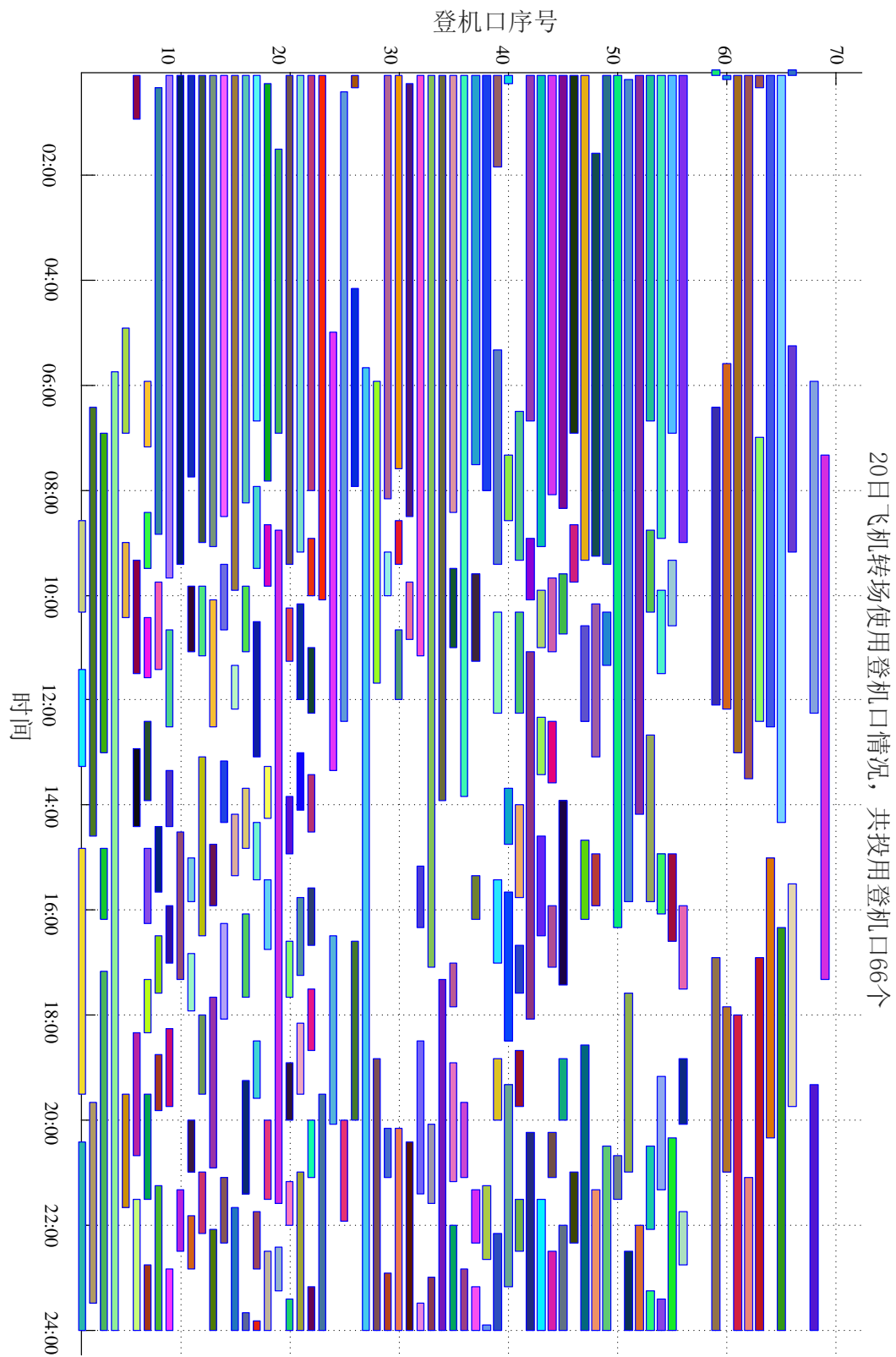


图 6.6 20 日全天各飞机起降时间流图

7 结论

将本文三个子问题的结果汇总，得到下述表格。

表 7.1 关于三个问题的成功分配到登机口的航班统计

成功分配到登机口的航班统计			
	航班总量	已分配航班数量	比例
问题 1	606	512	84.50%
问题 2	606	512	84.50%
问题 3	606	512	84.50%

成功分配到登机口的航班统计			
	窄航班总量	已分配窄航班数量	比例
问题 1	508	414	81.50%
问题 2	508	414	81.50%
问题 3	508	414	81.50%

成功分配到登机口的航班统计			
	宽航班总量	已分配宽航班数量	比例
问题 1	98	98	100%
问题 2	98	98	100%
问题 3	98	98	100%

从表 7.1 可以看出三个子问题均分配 512 个航班到固定登机口，均能优先保证最大化停靠固定登机口航班的数量。同时宽航班的分配比例也均为 100%，窄航班的分配比例均为 81.5%，说明三个子问题在航班-登机口分配的类型上存在着一致性。

表 7.2 关于三个问题的投运登机口数量统计

登机口投运数目			
	登机口总量	投运登机口总量	比例
问题 1	69	65	94.20%
问题 2	69	65	94.20%
问题 3	69	66	95.65%

登机口投运数目			
	航站楼登机口总量	航站楼投运登机口总量	比例
问题 1	28	27	96.42%
问题 2	28	28	100%
问题 3	28	28	100.00%

表 7.2 关于三个问题的投运登机口数量统计（续）

	登机口投运数目		
	卫星厅登机口总量	卫星厅投运登机口总量	比例
问题 1	38	41	92.70%
问题 2	41	37	90.20%
问题 3	41	38	92.68%

从表 7.2 可以看出三个子问题中 69 个登机口均没有全部用上，可能是由于登机口与航班类型之间的匹配问题导致部分登机口冗余，这部分登机口可作为备用登机口使用，因此最小化登机口的数量是有意义的。同时可以看出卫星厅登机口的利用率在 90%以上，可见新建卫星厅可以很大程度上解决机场登机口数量不足的问题。

表 7.3 关于三个问题的登机口平均使用率统计

	投运登机口平均使用率		
	所有投运飞机	航站楼所投运登机口	卫星厅所投运登机口
问题 1	60.86%	60.06%	61.43%
问题 2	60.85%	59.52%	61.85%
问题 3	60.28%	59.56%	60.81%

从表 7.3 可以看出登机口的平均使用率在 60%作用，考虑到航班之间存在 45 分钟的间隙，因此登机口的利用率较高，没有出现严重的登机口浪费的现象。而问题三的登机口平均使用率比问题一和问题二低，原因是问题三中投入了 66 个登机口，而问题一和问题二只需要使用 65 个登机口，可见如果没有优化登机口的数量，将会造成登机口的平均使用率大大下降，从而使得登机口的利用效率低下。

通过对上述结果的分析我们发现，卫星厅的建设可以缓解原有航站楼登机口不足的压力，但卫星厅的投运可能使游客的换乘时间增加。根据计算，新建卫星厅后，旅客的平均换乘紧张度为 0.297，没有换乘失败的旅客，因此卫星厅虽然在一定程度上增加了换乘时间，但旅客的平均换乘紧张度仍是一个较低的数字，同时并没有因为卫星厅的存在导致旅客出现换乘失败的现象，因此机场新增卫星厅对中转旅客影响在可以接受的范围内，卫星厅对于机场而言是利大于弊的。

8 模型的优缺点与展望

8.1 模型优点

1、本文在问题一中所建立的解析模型为整数规划模型，能够快速求得全局最优解。并且本文在问题一中采用逐次逼近的方法将多目标转化为约束形式，从而得到 Pareto 前沿以实现多目标全局最优。

2、对于涉及旅客换乘的登机口调度问题，解析法面临着“维数灾”问题而难以求解，基于种群的启发式算法只能将登机口时间约束转化为罚函数形式，导致难以收敛到正确的解空间中，而本文所采用的邻域搜索算法能在保证满足约束条件的前提下进行寻优，能够在有限的时间内求得合理的最优解。

3、本文所采用的邻域搜索算法是一种确定性算法，同一问题多次求解的结果必定相同，因此存在着很强的稳定性。

4、本文所采用的邻域搜索算法具有很高的扩展性，对于目标函数不同的问题只需简单的修改目标函数的表达式，而不需要重新构建整个程序。同时邻域搜索算法能很好的实现分阶段优化，适合求解多目标优化问题。

5、本文所采用的邻域搜索算法具有很强的鲁棒性，能够适应不同类型的优化目标，不要求目标函数连续可微。

8.2 模型确定

1、本文所提出的邻域搜索算法对算法起始点的敏感度高，算法起始点必须满足所有约束，并且影响着最终收敛值，较差的起始点可能会导致算法收敛至局部最优。

2、本文所提出的邻域搜索算法具有高次方时间复杂度，对内存的占用过高同时在求解速度上存在一定的劣势，不适合求解动态规划问题。

8.3 展望

1、参考元启发式算法，例如禁忌搜索算法，引入禁忌表，从而降低邻域搜索算法的时间复杂度，提高算法收敛速度。

2、本文由于没有考虑临时停机坪的后续处理，例如摆渡车等场景，同时没有考虑航班的延误情况，优化结果与现实存在较大的出入，因此在今后的工作中应当建立更加贴近实际的数学模型从而得到更具有现实意义的结论。

9 参考文献

- [1] Hemchand Kochukuttan and Sergey Shebalov, New Gate Planning Optimizer Perfects Gate Assignment Process, Ascend Contributors
- [2] Bouras A, Ghaleb M A, Suryahatmaja U S, et al. The Airport Gate Assignment Problem: A Survey[J]. The Scientific World Journal, 2014, (2014-11-20), 2014, 2014(6):9165-9172.
- [3] YALMIP, <https://yalmip.github.io/>, 20180918
- [4] GUROBI, <http://www.gurobi.com/>, 20180918
- [5] Xu J, Bailey G. The Airport Gate Assignment Problem: Mathematical Model and a Tabu Search Algorithm[M]// The airport gate assignment problem: mathematical model and a tabu search algorithm. 2001:3032-3032.
- [6] Ding H, Lim A, Zhu Y. New Heuristics for Over-Constrained Flight to Gate Assignments[J]. Journal of the Operational Research Society, 2004, 55(7):760-768.

10 附录

由于本文的程序代码与数据文件较多，故不将代码与数据文件置于附录，而是以附件的形式上传，本附录仅介绍附件中数据文件与程序的作用。

计算结果.csv 文件中包含有“问题一登机口”，“问题二登机口”和“问题三登机口”三个子问题的分配。

程序部分：

本文的程序均在 Intel Core i7-4710HQ 处理器下，8GB Ram，64 位 Windows 10 操作系统环境下运行。编程语言为 MATLAB 的 M 语言，所用 MATLAB 版本为 MATLAB 2014a，其中部分程序使用 YALMIP^[1]作为建模工具，并调用 GUROBI^[2]作为求解器。

[1] <https://yalmip.github.io/>

[2] <http://www.gurobi.com/>

第一问文件夹中：

solve1.m 是求解问题一中整数规划从而得到最大化有固定停机位的航班的程序。

solve1_2.m 是采用逐次逼近法得到 Pareto 前沿从而在最大化有固定停机位的航班的前提下确定最少需要的登机口的数量的程序。

第二问文件夹中：

program.m 为主程序，通过邻域搜索算法求解游客最小换乘手续时间。

program_station.m 为求解游客最小换乘手续时间后通过邻域搜索算法求解在最小换乘手续时间的基础上优化登机口的数量的程序。

youxiaohuancheng.m 为筛选 20 日当天有效换乘的函数。

fobj.m 为已知登机口分配后求得目标函数值的函数。

fcharu.m 为搜索“登机口内部航班平移”邻域的函数。

fjiaohuan.m 为搜索“登机口内部航班交换”邻域的函数。

ftingji.m 为搜索“登机口-临时机位航班交换”邻域的函数。

huancheng.m 为求得换乘乘客信息的函数。

isOK_jiaohuan.m 为检测搜索到的“登机口内部航班交换”是否为有效邻域的函数。

isOK_charu.m 为检测搜索到的“登机口内部航班平移”是否为有效邻域的函数。

isOK_tingji.m 为检测搜索到的“登机口-临时机位航班交换”是否为有效邻域的函数。

jiancha.m 为检测优化完成后得到的登机口分配方案是否满足所有约束的函数。

jisuan_charu.m 为计算“登机口内部航班平移”有效邻域对应的目标函数值的函数。

jisuan_jiaohuan.m 为计算“登机口内部航班交换”有效邻域对应的目标函数值的函数。

jisuan_tingji.m 为计算“登机口-临时机位航班交换”有效邻域对应的目标函数值的函数。

shijian.m 为已知登机口分配方案求取乘客换乘时间的函数。

第三问文件夹中：

program.m 为主程序，通过邻域搜索算法求解游客最小换乘紧张度。

program_station.m 为求解游客最小换乘紧张度后通过邻域搜索算法求解在最小换乘紧张度的基础上优化登机口的数量的程序。

youxiaohuancheng.m 为筛选 20 日当天有效换乘的函数。

fobj.m 为已知登机口分配后求得目标函数值的函数，以及输出换乘失败的信息。

fcharu.m 为搜索“登机口内部航班平移”邻域的函数。

fjiaohuan.m 为搜索“登机口内部航班交换”邻域的函数。

ftingji.m 为搜索“登机口-临时机位航班交换”邻域的函数。

huancheng.m 为求得换乘乘客信息的函数。

isOK_jiaohuan.m 为检测搜索到的“登机口内部航班交换”是否为有效邻域的函数。

isOK_charu.m 为检测搜索到的“登机口内部航班平移”是否为有效邻域的函数。

isOK_tingji.m 为检测搜索到的“登机口-临时机位航班交换”是否为有效邻域的函数。

jiancha.m 为检测优化完成后得到的登机口分配方案是否满足所有约束的函数。

jisuan_charu.m 为计算“登机口内部航班平移”有效邻域对应的目标函数值的函数。

jisuan_jiaohuan.m 为计算“登机口内部航班交换”有效邻域对应的目标函数值的函数。

jisuan_tingji.m 为计算“登机口-临时机位航班交换”有效邻域对应的目标函数值的函数。

shijian.m 为已知登机口分配方案求取乘客换乘时间的函数。